Introduction

CSE 410, Spring 2004
Computer Systems

http://www.cs.washington.edu/education/courses/410/04sp/
Reading and References

• Reading
  » Chapter 1, *Computer Organization and Design*, Patterson and Hennessy

• Other References
  » The Rope and Pulley Wonder, in *The Tinkertoy Computer*, A. K. Dewdney
Administrative

• Instructor:
  » Doug Johnson
  » djohnson@cs.washington.edu

• All class info is on the web site
  » http://www.cs.washington.edu/410
  » also known as
    • http://www.cs.washington.edu/education/courses/cse410/04sp/
Class Overview

• Provide an introduction to the inner workings of computer systems
• Levels of abstraction
  » bits, bytes, assembly language
  » operating system concepts
  » higher level languages - C, C++, Java, …
  » application programs
Goal

• You will understand
  » what is actually happening when a computer system is running application programs
• So that you will be able to
  » make good design choices as a developer, project manager, or system customer
  » calibrate your hype-o-meter with facts
The structure of this class

• The hardware / software interface
  » the elements of a computer system
  » what parts are visible to the software
  » instruction set architecture (ISA)

• Operating systems
  » services an OS performs for an application
  » design of various OS components
Elements of a computer system

• Start with a point of view
  » purchase a CD on the Web
  » get class schedule from MyUW
  » write a resume using Word
  » write a Java program to do image processing
  » write a C program to read real time data
  » write assembly language for matrix operations
  » write microcode for instruction emulation
“Top Level” elements

• At any level of abstraction, there are
  » elements at that level
  » the building blocks for those elements

• Rope analogy in the book
  » a cable: three hawsers twisted together
  » a hawser: three strands of many yarns
  » down to the molecular level and beyond
Purchase a CD on the Web

• the “top level” system includes
  » your browser, your desktop computer
  » connection to the internet (ISP)
  » server - http://www.amazon.com/
  » server application code
    • method="POST"
    • action="/exec/obidos/handle-buy-box=B00005NFZB/…"
    • ...

Write a resume using Word

• the “top level” system includes
  » winword.exe - the application program
  » Contemporary Resume.dot - document template
  » resume.doc - the file containing the text
  » Windows Explorer - file manager
  » network file and printer sharing
assembly language for matrix operations

• the “top level” system includes
  » programmer’s editor (eg, Context)
  » assembler - convert source to machine language
  » linker, loader - build and run executable
  » Instruction Set Architecture (ISA) that you are
    writing the code for
    • defines the programmer-visible face of the CPU
    • in this class, we will be writing for MIPS 1 ISA
Layers of abstraction

• Abstraction
  » isolates a layer from changes in the layer below
  » improves developer productivity by reducing detail needed to accomplish a task
  » helps define a single architecture that can be implemented with more than one organization
Architecture and Organization

• Architecture
  » defines elements and interfaces between layers
  » ISA: instructions, registers, addressing

• Organization
  » components and connections
  » how instructions are implemented in hardware
  » many different organizations can implement a single architecture
Computer Architecture

- Specification of how to program a specific computer family
  - what instructions are available?
  - how are the instructions formatted into bits?
  - how many registers and what is their function?
  - how is memory addressed?
- The MIPS 1 architecture is the basis for the first half of this course
Architecture Families

- IBM 360, 370, …
- PowerPC 601, 603, …
- DEC PDP-11
- Intel x86 286, 386, 486, Pentium, …
- Motorola 680x0
- MIPS R2000, R3000, R4000, R5000, …
Computer Organization

- Processor
  - datapath (functional units) manipulate the bits
  - control controls the manipulation

- Memory
  - cache memory - smaller, higher speed
  - main memory - larger, slower speed

- Input / Output
  - interface to the rest of the world
Organizations and Architectures

• Architecture is another abstraction layer
• One architecture can be implemented with many organizations
• One organization can support multiple architectures
• Different manufacturing technologies
  » TTL, ECL, PMOS, NMOS, CMOS
  » ropes and pulleys - see Dewdney reference
Many possible implementations

Figure 2.4 The Apraphulian AND gate.
A typical organization

main memory

processor

I/O bus

hard disk  floppy disk  CDROM drive  serial ports  network interface

processor/memory bus
Change Organization or Architecture?

• Theory
  » Organization changes provide incremental changes in speed and cost for same software
  » Architecture changes enable breakthrough changes in speed and cost for new software

• Real life
  » incremental changes are very rapid
  » breakthrough changes are very costly
A quick hardware tour

- System board
  - CPU, memory, I/O bus
- Hard disk
  - 3600+ RPM, 8ms latency, 3-15 ms seek
- Monitor
  - CRT, LCD
- Mouse, keyboard
  - embedded processors